A Construction Method of Interval-Valued Fuzzy Sets for Image Processing

A. Jurio, D. Paternain,
C. Lopez-Molina, H. Bustince
Dpto. de Automatica y Computacion
Universidad Publica de Navarra
31006 Pamplona, Spain
Email: aranzazu.jurio@unavarra.es

R. Mesiar
Dep. of Math. and Desc. Geometry
Slovak University of Technology
Bratislava, Slovakia
Inst. Information Theory & Automation
Academy of Sciences Czech Republic
CR-18208 Prague, Czech Republic
Email: mesiar@math.sk

G. Beliakov
Sch. of Information Technology
Deakin University
3125 Burwood, Australia
Email: gleb@deakin.edu.au

***Abstract*—In this work we present a new construction method of IVFSs from Fuzzy Sets. We use these IVFSs for image processing. Concretely, in this contribution we introduce a new image magnification algorithm using IVFSs. This algorithm is based on block expansion and it is characterized by its simplicity.**

***Index Terms*—Interval-Valued Fuzzy Sets, Image Processing, Image Magnification**

I. Introduction

In image processing problems we start from an image, that can be naturally interpreted as a Fuzzy Set. However, it has been proven that the use of Interval-valued Fuzzy Sets (IVFSs) in this kind of problems can provide some information about the neighbourhood of each pixel [5]. This is why in the specialized literature we can find numerous applications of IVFSs in image processing, to solve problems like filtering [2], edge detection [4] or segmentation [9], [17], [7].

To associate an IVFS with an image, in this work we present a new construction method of IVFSs from Fuzzy Sets, so every pixel in the image has an interval membership degree to the set. We interpret the length of each pixel's membership to the IVFS as a measure of the variation of intensities in the neighbourhood of that pixel. This is why the length of each interval is fixed beforehand.

Once the image is represented by an IVFS, we can apply any interval-valued method to it. In this work we focus on image scaling problem. There are two fundamental kinds of scaling: magnification, where the dimension of the image is enlarged [12], [15], [18], and reduction, that diminishes it [10]. In this contribution we work on image magnification, also called enlargement.

Image scaling is used in many applications. For instance, to upload images to a web page or to show images in devices such as screens, PDAs or mobile phones. Some of these devices have very limited memory. In these cases it is necessary to use simple image magnification algorithms.

There exist several techniques for image magnification [13]. Some of them are only based on one image, while others use several images in the magnification process. The most frequently used methods working with a single image are based on interpolation [1]. Common algorithms, such as nearest neighbour or bilinear interpolation, are computationally simple, but suffer from smudge problems, especially in the areas containing edges. Nevertheless, linear approximations are the most used ones since, even if they provide results worse than those obtained with cubic interpolation or splines, the computational cost of the latter is larger. Methods working with several images are also very common in video applications, to magnify a video sequence [16], [14]. Sets of images are also used to enlarge individual images in learning frameworks [8], [11].

We present a different approach for image magnification, based only on one image, constructed by blocks. In this sense, each area or block in the new image is obtained by a weighted aggregation of the intensities of a pixel and its neighbours in the original image.

This work is organized as follows: first, in Section II we recall some preliminary definitions. In Section III, we show the construction method of IVFSs. In Section IV, we describe in detail the image magnification algorithm. We finish the work with some experimental results in Section V and conclusions in Section VI.

II. Preliminaries

Let us denote by $L([0,1])$ the set of all closed subintervals in $[0,1]$, that is,

$$L([0,1]) = \{\mathbf{x} = [\underline{x}, \overline{x}] | (\underline{x}, \overline{x}) \in [0,1]^2 \text{ and } \underline{x} \leq \overline{x}\}.$$

$L([0,1])$ is a lattice with respect to the relation $\leq_L$, which is defined in the following way. Given $\mathbf{x}$,$\mathbf{y} \in L([0,1])$,

$$\mathbf{x} \leq_L \mathbf{y} \text{ if and only if } \underline{x} \leq \underline{y} \text{ and } \overline{x} \leq \overline{y}.$$

The relation above is transitive, antisymmetric and it expresses the fact that $\mathbf{x}$ strongly links to $\mathbf{y}$, so that $(L([0,1]), \leq_L)$ is a complete lattice, where the smallest element is $0_L = [0,0]$, and the largest is $1_L = [1,1]$.

Definition 1: An interval-valued fuzzy set A on the universe $U \neq \emptyset$ is a mapping $A : U \rightarrow L([0,1])$.

978-1-61284-076-5/11/$26.00 ©2011 IEEE

We denote by $IVFSs(U)$ the set of all IVFSs on U. Similarly, $FSs(U)$ is the set of all fuzzy sets on U.

From now on, we denote by $W([\underline{x}, \overline{x}])$ the length of the interval $[\underline{x}, \overline{x}]$; that is, $W([\underline{x}, \overline{x}]) = \overline{x} - \underline{x}$.

Definition 2: Let $\alpha \in [0, 1]$. The operator $K_\alpha : L([0, 1]) \to [0, 1]$ is defined as a convex combination of the bounds of its argument, i.e.

$$K_\alpha(\mathbf{x}) = \underline{x} + \alpha(\overline{x} - \underline{x})$$

for all $\mathbf{x} \in L([0, 1])$.

Clearly, the following properties hold:

1) $K_0(\mathbf{x}) = \underline{x}$ for all $\mathbf{x} \in L([0, 1])$,
2) $K_1(\mathbf{x}) = \overline{x}$ for all $\mathbf{x} \in L([0, 1])$,
3) $K_\alpha(\mathbf{x}) = K_\alpha([K_0(\mathbf{x}), K_1(\mathbf{x})]) = K_0(\mathbf{x}) + \alpha(K_1(\mathbf{x}) - K_0(\mathbf{x}))$ for all $\mathbf{x} \in L([0, 1])$.

Let $A \in IVFSs(U)$ and $\alpha \in [0, 1]$. Then, we denote by $K_\alpha(A)$ the fuzzy set

$$K_\alpha(A) = \{u_i, K_\alpha(A(u_i))|u_i \in U\}.$$

Proposition 1: For all $\alpha, \beta \in [0, 1]$ and $A, B \in IVFSs(U)$, it is verified that

(a) If $\alpha \leq \beta$, then $K_\alpha(A) \leq K_\beta(A)$.
(b) If $A \leq_L B$ then $K_\alpha(A) \leq K_\alpha(B)$.

where $\leq$ is Zadeh's order relation.

III. Construction of Interval-Valued Fuzzy Sets of Fixed Length

In this section we propose a construction method of elements in $L([0, 1])$ such that their length is fixed beforehand. As we have already mentioned, this fact is very important because we want to represent the variation of the intensities around each pixel by the length of the interval. Besides, this variation is adjusted by δ, a scaling factor.

Proposition 2: The mapping $F : [0, 1]^2 \times [0, 1] \to L([0, 1])$ given by

$$F(x, y, \delta) = [\underline{F}(x, y, \delta), \overline{F}(x, y, \delta)]$$

where

$$\underline{F}(x, y, \delta) = x(1 - \delta y)$$

$$\overline{F}(x, y, \delta) = x(1 - \delta y) + \delta y$$

satisfies that:

1) $\underline{F}(x, y, \delta) \leq x \leq \overline{F}(x, y, \delta)$ for all $x \in [0, 1]$;
2) $F(x, 0, \delta) = [x, x]$;
3) $F(0, y, \delta) = [0, \delta y]$;
4) $F(x, y, 0) = [x, x]$;
5) $W(F(x, y, \delta)) = \delta y$.
6) If $y_1 \leq y_2$ then $W(F(x, y_1, \delta)) \leq W(F(x, y_2, \delta))$ for all $x, \delta \in [0, 1]$;

Theorem 1: Let $A_F \in FSs(U)$ and let $\omega, \delta : U \to [0, 1]$ be two mappings. Then

$$A = \{(u_i, A(u_i) = F(\mu_{A_F}(u_i), \omega(u_i), \delta(u_i)))|u_i \in U\}$$

is an Interval-Valued Fuzzy Set.

Corollary 1: In the setting of Theorem 1, if for every $u_i \in U$ we take $\delta(u_i) = 1$ then

$$\omega(u_i) = W(F(\mu_{A_F}(u_i), \omega(u_i), 1)).$$

Notice that under the conditions of Corollary 1 the set A is given as follows:

$$A = \{(u_i, \mu_{A_F}(u_i)(1 - \omega(u_i)), \mu_{A_F}(u_i)(1 - \omega(u_i)) + \omega(u_i))|u_i \in U\}$$

Example 1: Let $U = \{u_1, u_2, u_3, u_4\}$ and let $A_F \in FSs(U)$ given by

$$A_F = \{(u_1, 0.3), (u_2, 1), (u_3, 0.5), (u_4, 0.8)\}$$

and $\omega(u_i) = 0.3, \delta(u_i) = 1$ for all $u_i \in U$. By Corollary 1 we obtain the following Interval-Valued Fuzzy Set:

$$A = \{(u_1, [0.21, 0.51]), (u_2, [0.7, 1.00]), (u_3, [0.35, 0.65]), (u_4, [0.56, 0.86])\}$$

IV. Image Magnification Algorithm

In this section we propose a gray-scale image magnification algorithm that uses IVFSs and the K_α operators.

In this work, we consider an image Q of $N \times M$ pixels as a set of $N \times M$ elements arranged in rows and columns. Therefore, we consider an image as a $N \times M$ matrix. We denote by q_{ij} the intensity of the pixel at the position (i, j) of the Q matrix, with $i \in \{1, \ldots, N\}$, $j \in \{1, \ldots, M\}$. In this contribution we work with gray-scale images whose intensities take values in the interval [0,255]. We normalize them in order to have values in $[0, 1]$.

The purpose of our algorithm is, given an image Q of dimension $N \times M$, to magnify it $2n+1$ times; that is, to build a new image of dimension $N' \times M'$ with $N' = (2n+1) \times N$, $M' = (2n + 1) \times M$, $n \in \mathbb{N} - \{0\}$ with $2n + 1 \leq N$ and $2n + 1 \leq M$.

The algorithm consists of the following steps:

1. Take $\delta \in [0, 1]$.
2. FOR each pixel in position (i, j) DO
   - 2.1. Fix a grid V of dimension $(2n + 1) \times (2n + 1)$ centered at (i, j).
   - 2.2. Calculate W as the difference between the largest and the smallest intensities of the pixels in V.
   - 2.3. Build the interval $F(q_{ij}, W, \delta)$.
   - 2.4. Build a block V' equal to V.
   - 2.5. FOR each element (k, l) of V' DO

     $$q_{kl} = K_{q_{kl}}(F(q_{ij}, W, \delta)).$$

     ENDFOR

   ENDFOR

Algorithm 1

We explain the steps of Algorithm 1 through an intuitive example. Given the image in Figure 1 of dimension 5×5, we want to build a magnified image of dimension 15×15 ($n = 1$).

0.7	0.6	0.65	0.4	0.41
0.66	0.59	0.6	0.5	0.46
0.62	0.7	0.6	0.52	0.48
0.6	0.68	0.55	0.53	0.5
0.63	0.62	0.52	0.5	0.5

Fig. 1. Example: original image

Step 1. Take $\delta \in [0, 1]$. In the example we take $\delta = 1$.

Step 2.1. Fix a grid V of dimension $(2n + 1) \times (2n + 1)$ centered at each pixel. This grid represents the neighbourhood that is used in the magnification of each pixel from the image. The intensities of the pixels in this grid provide the information to get the length of the membership interval built through F. In the example, for pixel $(2, 3)$ (marked in dark gray in Figure 2), we fix a grid of dimension 3×3 around it (in light gray).

0.7	0.6	0.65	0.4	0.41
0.66	0.59	0.6	0.5	0.46
0.62	0.7	0.6	0.52	0.48
0.6	0.68	0.55	0.53	0.5
0.63	0.62	0.52	0.5	0.5

Fig. 2. Example: Grid V in original image

Remark. For pixels in the first or the last row/column, we choose a grid centered at them as shown in Figure 3.

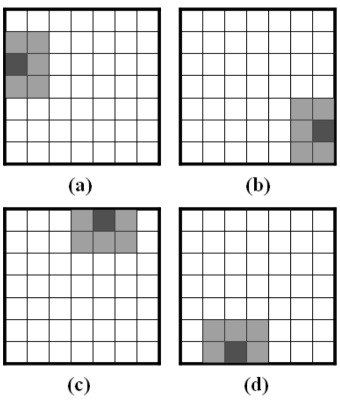

Fig. 3. Example: grid for pixels in the first or last row/column. (a) Pixel in the first column, (b) Pixel in the last column, (c) Pixel in the first row, (d) Pixel in the last row.

Step 2.2. Calculate W as the difference between the largest and the smallest of the intensities of the pixels in V. For pixel $(2, 3)$ we calculate W as:

$$\begin{aligned} W = &\max(0.6, 0.65, 0.4, 0.59, 0.6, 0.5, 0.7, 0.6, 0.52) - \\ &\min(0.6, 0.65, 0.4, 0.59, 0.6, 0.5, 0.7, 0.6, 0.52) = \\ &= 0.7 - 0.4 = 0.3 \end{aligned}$$

Step 2.3. Build the interval $\mathbf{F}(\mathbf{q_{ij}}, \mathbf{W}, \delta)$. We associate to each pixel an interval of length $\delta \cdot W$ following the method explained in Section III:

$$F(q_{ij}, W, \delta) = [q_{ij}(1 - \delta \cdot W), q_{ij}(1 - \delta \cdot W) + \delta \cdot W].$$

In the example, the interval associated to pixel $(2, 3)$ is given by:

$$F(0.6, 0.3, 1) = [0.6(1-0.3), 0.6(1-0.3)+0.3] = [0.42, 0.72].$$

Step 2.4. Build a block $\mathbf{V'}$ equal to $\mathbf{V}$. This new block is shown in Figure 4.

0.6	0.65	0.4
0.59	0.6	0.5
0.7	0.6	0.52

Fig. 4. Copied V' block for pixel $(2, 3)$.

Step 2.5. Calculate $\mathbf{K_{q_{kl}}(F(q_{ij}, W, \delta))}$ for each pixel. We are going to expand each pixel (i, j) in image Q over the new block V'. In the example, the block V' associated to pixel $(2, 3)$ is expanded as shown in Figure 5.

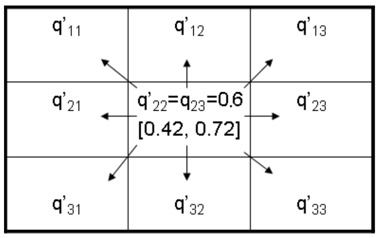

Fig. 5. Expanded block for pixel q_{23}

In this step, we are going to use the result of Proposition 3.

Proposition 3: In the settings of Proposition 1, if we take $\alpha = x$, then

$$K_x(F(x, y, \delta)) = x$$

for all $x, y, \delta \in [0, 1]$.

Proof. $K_x(F(x, y, \delta)) = K_x([x(1 - \delta y), x(1 - \delta y) + \delta y]) = x(1 - \delta y) + x \cdot W(F(x, y \cdot \delta))$.

From condition 5) in Proposition 2

$K_x(F(x, y, \delta)) = x(1 - \delta y) + x\delta y = x$ □

To keep the value of the original pixel at the center of the new block, Proposition 3 states that α should be equal to the intensity of that pixel. In the case of pixel $(2, 3)$ we have

$$0.6 = q'_{22} = K_{q_{23}}([0.42, 0.72]) = 0.42 + q_{23}0.3 = 0.6.$$

We extend this method to fill in the remaining pixels in the block. In this way, from Proposition 3 we take α as the intensity of each pixel in the original block V' :

- $\alpha = q_{12}$. Then $q'_{11} = 0.42 + q_{12}0.3 = 0.42 + 0.6 \cdot 0.3 = 0.6$
- $\alpha = q_{13}$. Then $q'_{12} = 0.42 + q_{13}0.3 = 0.42 + 0.65 \cdot 0.3 = 0.615$
- $\alpha = q_{14}$. Then $q'_{13} = 0.42 + q_{14}0.3 = 0.42 + 0.4 \cdot 0.3 = 0.54$
- $\cdots$
- $\alpha = q_{34}$. Then $q'_{33} = 0.42 + q_{34}0.3 = 0.42 + 0.52 \cdot 0.3 = 0.576$

In Figure 6 we show the expanded block for pixel $(2, 3)$ in the example.

0.6	0.615	0.54
0.597	0.6	0.57
0.63	0.6	0.576

Fig. 6. Expanded block for pixel q_{23}

Once each of the pixels has been expanded, we join all the blocks to create the magnified image, as shown in Figure 7.

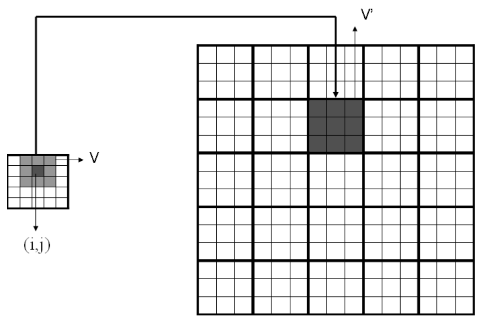

Fig. 7. Magnified image

V. Experimental Results

In this section we apply our approach in four images and we study the results. We follow these steps (see Figure 8):

1. We start from images of 255×255 and we reduce them to a 85×85 size using the reduction algorithm proposed in [6].
2. Next, with Algorithm 1, we magnify the images to a 255×255 size.
3. Finally, we compare the images that we obtain with the original ones.

In order to compare the images we use the measures developed in [3]. Concretely, we use the expression

$$S(Q, Q') = \frac{1}{N \times M} \sum_{i=1}^{N} \sum_{j=1}^{M} 1 - |q_{ij} - q'_{ij}| \qquad (1)$$

TABLE I
Comparison of the reconstructed images with the original one

	$\delta = 0$	$\delta = 0.25$	$\delta = 0.5$	$\delta = 0.75$	$\delta = 1$
Ship	0.9596	0.9605	0.9610	0.9609	0.9606
Church	0.9718	0.9730	0.9732	0.9730	0.9723
Cows	0.9324	0.9335	0.9341	0.9342	0.9338
Car	0.9506	0.9511	0.9513	0.9512	0.9508

where Q, Q' are two images of dimension $N \times M$.

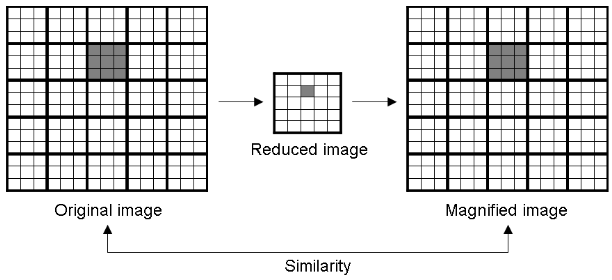

Fig. 8. Schema to check the algorithm

Figure 9 depicts the original images (first column) and their reductions (second column).

A. Initial Study of the Values of δ

In this experiment we are going to study the effect of the parameter δ in Algorithm 1. To do so, we start executing it over the previous images with values $\delta = \{0, 0.25, 0.5, 0.75, 1\}$.

When $\delta = 0$, we know by Proposition 2 that

$$F(x, y, 0) = [x, x]$$

for all $x, y \in [0, 1]$, and we aslo know that

$$K_\alpha([x, x]) = x$$

for all $\alpha \in [0, 1]$. In this way, when we apply the magnification method taking $\delta = 0$ we build blocks in which all the elements take the value of the central pixel (see Figure 10).

In this sense, when $\delta = 0$ we lose information from the neighbourhood, that is, the reconstructed blocks do not keep the same relation than the original pixel with the surrounding ones. The application of this value for the parameter δ leads to poor solutions, as we can see in Figure 10.

Next, we analyze the remaining cases: $\delta = 0.25$, $\delta = 0.5$, $\delta = 0.75$ and $\delta = 1$. In Figure 11 we show all the test images magnifications for the chosen values of δ.

We must point out that when δ increases, the length of the interval associated with each pixel increases too. In this way, it also increases the range in which the intensities of pixels in each reconstructed block vary. Observe that if we take $\delta = 1$ the quality of areas around edges diminishes.

To compare the obtained images with the original one we use the comparison index given in Equation 1. In Table I we show the results. We observe that the best solutions are obtained when we take intermediate values of δ, that is, values of δ close to 0.5.

Fig. 11. Reconstructed images with different values of the parameter δ

B. Further Study of the Values of δ

Next we carry on a deeper analysis of our study. Up to now, we have worked with 5 values of δ uniformly distributed between 0 and 1. Now, we refine our study working with 100 values of δ uniformly distributed between 0 and 1. In Figures 12, 13, 14 and 15 we show the accuracy of our solutions. In the abscissa axis we show the values of parameter δ, while in the ordinate axis we show the similarities between the reconstructed images and the original ones.

From the resulting graphs we observe that the best results are obtained for $\delta = 0.59$ for Ship image, $\delta = 0.49$ for Church image, $\delta = 0.66$ for Cows image and $\delta = 0.55$ for Car image. We also observe that reconstructions loose quality as δ tends to zero or one, as it has been experimentally shown previously.

VI. Conclusion

In this work we have introduced a new construction method of IVFSs starting from fuzzy sets. The importance of this construction is that the length of the constructed interval is fixed a priori. It is used in the application we have presented, where represents the variation of intensities around each pixel. This application uses the construction method to develop a new image magnification algorithm. Instead of most of the published methods, this algorithm is not based on interpolation. In this way, a new block is constructed for every pixel

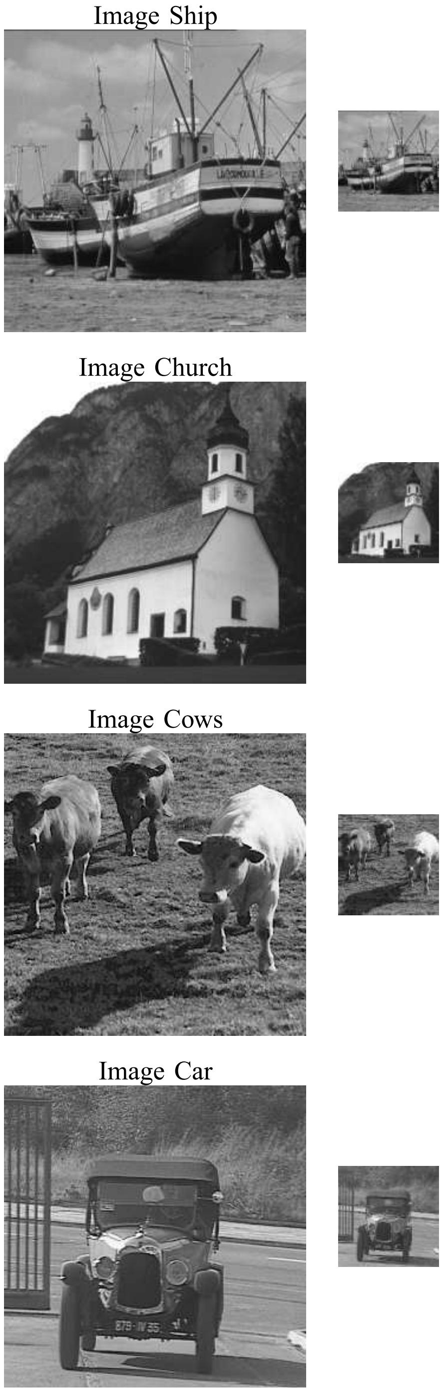

Fig. 9. Original images and reductions

Fig. 10. Images built with $\delta = 0$

of the image, and the central pixel of that block maintains the intensity of the original pixel. To fill in the rest of the pixels, we have used the relation between the pixel in the original image and its neighbours. The parametrization used in the algorithm allows to adapt it in order to look for the optimal set-up for each image. We have also studied this parameter, concluding that the best value is always far away from the bounds of its domain (0 and 1). Moreover, we must stress the simplicity of this approach with respect to other methods published in the specialized literature.

Acknowledgment

This paper has been partially supported by the National Science Foundation of Spain, reference TIN2010-15055, by the Research Services of the Universidad Publica de Navarra and by the grants P402/11/0378 and APVV-0012-07.

References

[1] A. Amanatiadis, I. Andreadis, *A survey on evaluation methods for image interpolation*, Measurement Science & Technology, 20 (2009) 104015.

[2] A. Bigand, O. Colot, *Fuzzy filter based on interval-valued fuzzy sets for image filtering*, Fuzzy Sets and Systems 161(2010)96-117.

[3] H. Bustince, M. Pagola, E. Barrenechea, Construction of fuzzy indices from DI-subsethood measures: Application to the global comparison of images, Information Sciences, 177 (2007) 906-929.

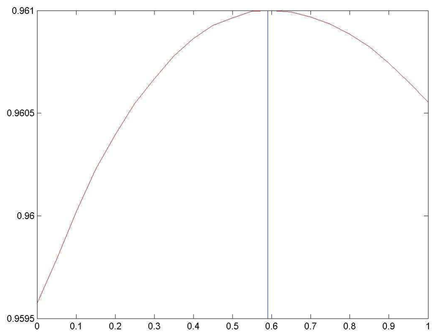

Fig. 12. Comparison of Ship image reconstructed with different δ

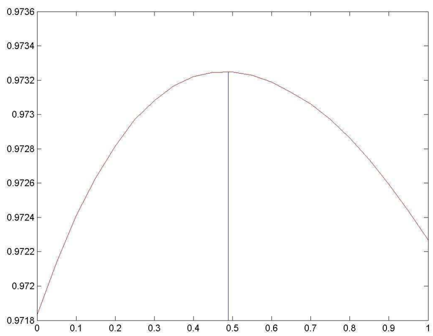

Fig. 13. Comparison of Church image reconstructed with different δ

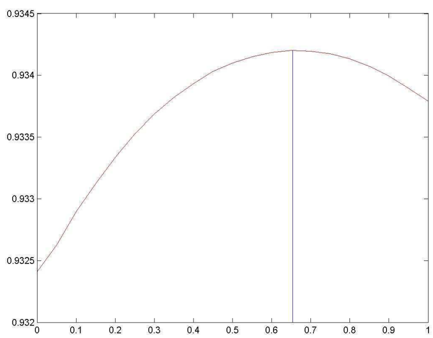

Fig. 14. Comparison of Cows image reconstructed with different δ

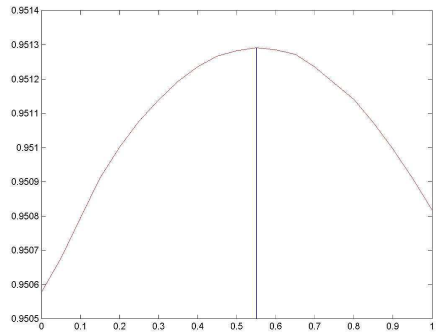

Fig. 15. Comparison of Car image reconstructed with different δ

[4] H. Bustince, E.Barrenechea, M.Pagola, J.Fernandez, *Interval-valued fuzzy sets constructed from matrices: Application to edge detection*, Fuzzy Sets and Systems 160 (2009) 1819-1840.

[5] H. Bustince, E. Barrenechea, J. Fernandez, M. Pagola, J. Montero, C. Guerra, *Contrast of a fuzzy relation*, Information Sciences 180 (2010) 1326-1344.

[6] H. Bustince, D. Paternain, B. De Baets, T. Calvo, J. Fodor, R. Mesiar, J. Montero, A. Pradera, *Two methods for image compression/reconstruction using OWA operators*, Recent Developments in the Ordered Weighted Averaging Operators: Theory and Practice, Springer, 2011, in press.

[7] H. Bustince, E. Barrenechea, M. Pagola, J. Fernandez, J. Sanz, *Comment on: "Image thresholding using type II fuzzy sets". Importance of this method*, Pattern Recognition 43 (2010) 3188–3192.

[8] Prakash P. Gajjar, Manjunath V. Joshi, *New Learning Based Super-Resolution: Use of DWT and IGMRF Prior*, IEEE Transactions on Image Processing 19 (2010) 1201-1213.

[9] A. Jurio, M. Pagola, D. Paternain, C. Lopez-Molina, P. Melo-Pinto, *Interval-valued restricted equivalence functions applied on Clustering Techniques*, in 13th International Fuzzy Systems Association World Congress and 6th European Society for Fuzzy Logic and Technology Conference 2009, Portugal.

[10] E. Karabassis, M.E. Spetsakis, *An analysis of image interpolation, defferentiation, and reduction using local polynomial*, Graphical Models and image processing 57 (1995) 183-196.

[11] K.S. Ni, T.Q. Nguyen, *Image Superresolution Using Support Vector Regression*, IEEE Transactions on Image Processing 16 (2007) 1596-1610.

[12] S.C. Park, M.K. Park, M.G. Kang, *Super-resolution image reconstruction: A technical overview*, IEEE Signal Processing Magazine, 20 (2003) 21-36.

[13] I. Perfilieva, *Fuzzy transforms: Theory and applications*, Fuzzy Sets and Systems 157 (2006) 993 1023.

[14] M. Protter, M. Elad, *Super Resolution With Probabilistic Motion Estimation*, IEEE Transactions on Image Processing 18 (2009) 1899-1904.

[15] G. Qiu, *Interresolution Look-up Table for Improved Spatial Magnification of Image*, Journal of Visual Communication and Image Representation, 11 (2000) 360-373.

[16] H. Takeda, P. Milanfar, M. Protter, M. Elad, *Super-Resolution Without Explicit Subpixel Motion Estimation*, IEEE Transactions on Image Processing 18 (2009) 1958-1975.

[17] H.R. Tizhoosh, *Image thresholding using type II fuzzy sets*, Pattern Recognition 38 (2005) 2363-2372.

[18] M. Unser, A. Aldroubi and M. Eden, *Enlargement and reduction of digital images with minimum loss of information*, IEEE Transactions on Image Processing 4 (1995) 247-258, 1995.